

CLAIM AMENDMENTS:

The following listing of claims will replace all prior versions and listings of the claims.

TPE APPARATUS

1. (Currently Amended) A telecommunication protocol engine device comprising a ~~microprocessor linked to a flash memory, wherein:~~
~~the flash~~ a first memory ~~comprises a first telecommunication protocol expressed as a first~~
~~template comprising one or more~~ storing one or more telecommunication protocol
templates, each of the one or more telecommunication protocol templates
including a plurality of virtual machine instructions to implement a
telecommunication protocol and to implement a finite state machine;
a second memory to store state data related to the finite state machine;
~~the microprocessor comprises a CPU linked to a random access memory (RAM) and a~~
processor having access to the first memory, the second memory and firmware,
the firmware adapted to operate a virtual machine, and wherein the CPU is
adapted to processor:
direct causes the virtual machine to read the first template of first virtual machine
instructions from the flash first memory, the first virtual machine
instructions selected from a first telecommunication protocol template to
implement a first telecommunication protocol;
~~store a current~~ stores the first template virtual machine instruction instructions in
the RAM second memory;
~~receive~~ receives first template state data to initialize the finite state machine; and
~~execute~~ executes the current first template virtual machine instruction instructions
using the first template state data and the finite state machine.
2. (Currently Amended) The telecommunication protocol engine device of claim 1,
wherein the first telecommunication protocol is selected from the group consisting of includes at
least one of a Session Initiation Protocol (SIP), a H.323 protocol, a STUN protocol, and a
Dynamic Host Configuration Protocol (DHCP).

3. (Canceled).

4. (Currently Amended) The telecommunication protocol engine device of claim 1 further comprising an I/O port, and wherein the processor ~~CPU is further adapted to receive~~ receives the first template state data ~~from~~ via the I/O port.

5. (Currently Amended) The telecommunication protocol engine device of claim 1, wherein the flash first memory further ~~comprising~~ includes second virtual machine instructions selected from a second telecommunication protocol ~~expressed as a second template to implement~~ a second telecommunication protocol comprising one or more virtual machine instructions, and wherein the CPU is further adapted to:

~~direct the virtual machine to read the second template of virtual machine instructions~~
~~from the flash memory;~~
~~store a current second template virtual machine instruction in the RAM;~~
~~receive second template state data; and~~
~~execute the current second template virtual machine instruction using the second template~~
~~state data.~~

6. (Currently Amended) The telecommunication protocol engine device of claim 5, wherein the second telecommunication protocol ~~is selected from the group consisting of~~ includes at least one of a Session Initiation Protocol (SIP), a H.323 protocol, a STUN protocol, and a Dynamic Host Configuration Protocol (DHCP).

7. (Currently Amended) The telecommunication protocol engine device of claim 5, wherein the second telecommunication protocol ~~template of virtual machine instructions~~ comprises ~~includes~~ one or more second template finite state machines.

8 -10. (Canceled).

11. (Currently Amended) The telecommunication protocol engine device of claim 1, wherein the ~~microprocessor~~ processor is linked to the flash first memory via a data bus.

12. (Currently Amended) The telecommunication protocol engine device of claim 11, wherein the data bus is a serial bus.

13. (Currently Amended) The telecommunication protocol engine device of claim 11, wherein the data bus is a parallel bus.

14. (Currently Amended) A method of implementing a telecommunication protocol ~~using a telecommunication protocol engine~~, the method comprising:

receiving a call at a ~~microprocessor~~ processor to implement a first telecommunication protocol;

selecting a first telecommunication protocol template, wherein the first

telecommunication protocol template ~~comprises~~ includes one or more virtual machine instructions to implement the first telecommunication protocol expressed as one or more virtual machine instructions;

~~directing~~ causing a virtual machine to read a first virtual machine instruction of the first telecommunication protocol template from a first memory;

initializing ~~the virtual machine~~ with a first finite state machine defined by the first telecommunication protocol template using first template state data to arrive at a first ~~current~~ template state; and

storing updated first template state data in a second memory, wherein the updated first template state data is determined based on the first finite state machine.

~~implementing the first telecommunication protocol; and~~

~~entering a new first state.~~

15. (Currently Amended) The method of ~~implementing a telecommunication protocol~~ using a telecommunication protocol engine of claim 14, wherein the first telecommunication protocol is selected from the group consisting of includes one of a Session Initiation Protocol (SIP), a H.323 protocol, and a STUN, and a dynamic host configuration protocol (DHCP).

16. (Canceled).

17. (Currently Amended) The method of ~~implementing a telecommunication protocol~~ using a telecommunication protocol engine of claim 14, wherein the first template state data is received by ~~[[the]]~~ a telecommunication protocol engine.

18. (Currently Amended) The method of ~~implementing a telecommunication protocol using a telecommunication protocol engine~~ of claim 14, wherein the method further comprises: receiving a call at a ~~microprocessor~~ the processor to implement a second telecommunication protocol;
selecting a second telecommunication protocol template, wherein the second telecommunication protocol template includes ~~comprises a second telecommunication protocol expressed as one or more virtual machine instructions to implement the second telecommunication protocol~~;
~~directing~~ causing the virtual machine to read a first virtual machine instruction of the second telecommunication protocol template from the first memory;
initializing ~~the virtual machine with a second finite state machine defined by the second telecommunication protocol template using second template state data to arrive at a second current template state; and~~
storing updated second template state data in the second memory, wherein the updated second template state data is determined based on the second finite state machine.
~~implementing the second telecommunication protocol; and~~
~~entering a new second state.~~

19. (Currently Amended) The method of ~~implementing a telecommunication protocol using a telecommunication protocol engine~~, of claim 18, wherein the second telecommunication protocol is selected from the group consisting of includes one of a Session Initiation Protocol (SIP), a H.323 protocol, and a STUN, and a dynamic host configuration protocol (DHCP).

20. (Canceled).

21. (Currently Amended) The method of ~~implementing a telecommunication protocol using a telecommunication protocol engine~~ of claim 18, wherein second template state data is received by ~~[[the]]~~ a telecommunication protocol engine.

22. (Currently Amended) The method of ~~implementing a telecommunication protocol using a telecommunication protocol engine~~ of claim 18, wherein the first template state data is determined based on the second finite state machine received from at least one of the one or more second template finite state machines.

23. (Currently Amended) The method of ~~implementing a telecommunication protocol using a telecommunication protocol engine~~ of claim 18, wherein the second template state data is determined based on the first finite state machine received from at least one of the one or more first template finite state machines.

24-43. (Canceled).

44. (New) The telecommunication protocol engine device of claim 1, wherein execution of the first virtual machine instructions changes the first template state data based on the finite state machine, and wherein the processor stores the changed first template state data at the second memory.

45. (New) The telecommunication protocol engine device of claim 44, wherein the processor receives a subsequent virtual machine instruction of the first telecommunication protocol template from the first memory based on the changed first template state data and the finite state machine.

46. (New) The telecommunication protocol engine device of claim 44, wherein the second memory does not store an entire set of virtual machine instructions from the first telecommunication protocol template at a single time.

47. (New) The telecommunication protocol engine device of claim 1, wherein the first memory has a larger storage capacity than the second memory.

48. (New) The telecommunication protocol engine device of claim 1, wherein the first memory includes a FLASH memory device.

49. (New) The telecommunication protocol engine device of claim 48, wherein the second memory includes a RAM memory device.

50. (New) The telecommunication protocol engine device of claim 5, wherein the processor:

causes the virtual machine to read the second virtual machine instructions from the first memory in response to a request to implement the second telecommunication protocol;

stores the second virtual machine instructions in the second memory;

receives second template state data related to initializing a second finite state machine related to the second telecommunication protocol; and

executes the second virtual machine instructions using the second template state data and the second finite state machine.

51. (New) The telecommunication protocol engine device of claim 50, wherein the second template state data is determined based at least partially on the first template state data.

52. (New) The method of claim 14, further comprising causing the virtual machine to read a subsequent first virtual machine instruction of the first telecommunication protocol template from the first memory based on the updated first template state data and the first finite state machine.

53. (New) The method of claim 52, wherein no more than one virtual machine instruction to implement the first telecommunication protocol is stored in the second memory at any time.

54. (New) The method of claim 14, further comprising:

causing the virtual machine to read a second virtual machine instruction of the first telecommunication protocol template from the first memory based on the updated first template state data and the first finite state machine;

using the first finite state machine to determine a second template state; and

storing the second template state data in the second memory, wherein the updated first template state data is determined based on the first finite state machine.

55. (New) The method of claim 54, wherein the first virtual machine instruction and the second virtual machine instruction are not stored in the second memory concurrently.